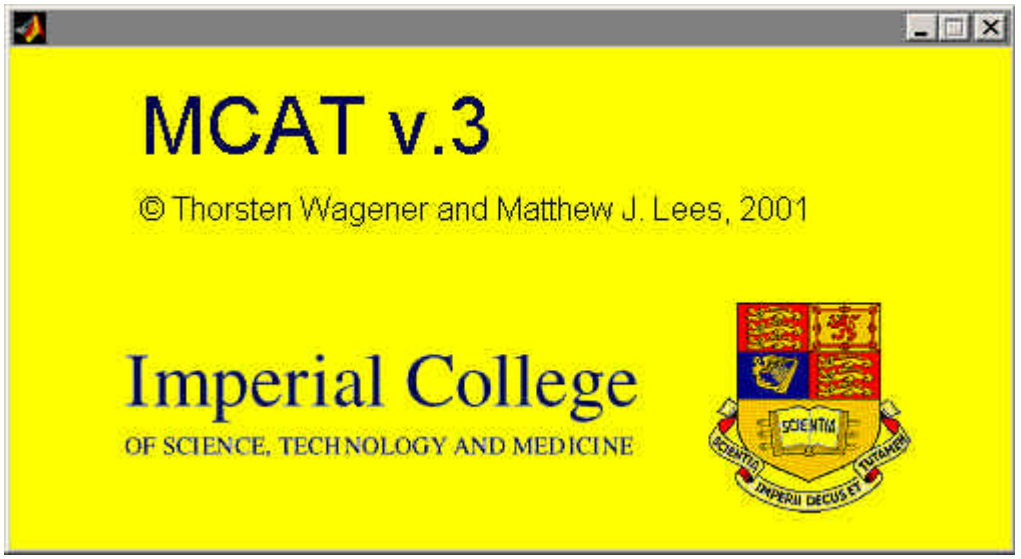


MONTE-CARLO ANALYSIS TOOLBOX USER MANUAL

Thorsten Wagener
Matthew J. Lees
Howard S. Wheater

Civil and Environmental Engineering Department
Imperial College of Science Technology and Medicine
London, SW7 2BU, UK

http://ewre.cv.ic.ac.uk/

CONTENTS

1

INTRODUCTION

THE TARGET GROUP OF THIS MANUAL

This manual was written for users of the Monte Carlo Analysis Toolbox (MCAT). Its emphasis lies on application of the toolbox. Only the part of the toolbox visible to the user is described. No information about the programming code is given.

OBJECTIVES OF THE MCAT

The Monte-Carlo Analysis Toolbox (MCAT) is a collection of MATLAB analysis and visualisation functions integrated through a graphical user interface. MATLAB is a numerical programming and visualisation package: see *www.mathworks.com* for more details and for information on availability. The toolbox can be used to analyse the results from Monte-Carlo (MC) parameter sampling experiments or from model optimisation methods that are based on population evolution techniques, for example, genetic algorithms (Goldberg, 1989). A number of powerful techniques are included in the toolbox to investigate the structure, sensitivity, and parameter and output uncertainty of mathematical models. Although this toolbox has been developed within the context of ongoing hydrological research, all functions can be used to investigate any dynamic mathematical model.

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE

- Any computer able to run Matlab version 5.2 or higher (Mathworks, 1996).

SOFTWARE

- Any platform supporting Matlab.
- Matlab version 5.2 or higher (RRMT is developed in Matlab 5.2, it might therefore not work properly in older versions).

ASSUMPTIONS

It is assumed that the user has a basic knowledge of the Matlab programming environment. Explanations about how to use Matlab are kept to a minimum within this manual.
The user is referred to the help documentation delivered with every Matlab version or to the Mathworks webside (http:\\www.mathworks.com) for more detailed information. A large number of books about Matlab is also available. Two examples are Etter, D.M. 1997. Engineering problem solving with Matlab - 2nd Edition. Prentice-Hall, USA (for a first introduction) and Hanselman, D. and Littlefield, B. 1998. Mastering Matlab 5 - A comprehensive tutorial and reference. Prentice-Hall, USA (for the more advanced user).

BACKGROUND

Many of the tools available in the MCAT aim to help investigate the identifiability of a model, where a well identified model is seen as a model with behavioural (possible, feasible) parameters that lie within a relatively small

region of the parameter space. Assessment of the identifiability of a model is important if the model is to be used for scientific hypothesis testing and uncertainty analysis, for example where parameters are given physical meaning and used to interpret the nature of a system, and to predict future behaviour, including impacts of changes to the system. Over the last 30 years it has become apparent that many popular models suffer from a lack of identifiability (e.g. Johnston and Pilgrim, 1976; Mein and Brown, 1978; Franchini and Pacciani, 1991). This can result in feasible model parameterisations over a wide range of the parameter space which produce equally or near equally good fits to the observed data.

A major cause of model identifiability problems is that, since internal model states are not observed, there are often many different combinations of internal states that combine to produce similar outputs. Improved identifiability in this situation is likely to be achieved through the availability of additional data (see for example Franks *et al*., 1998). An investigation of the identifiability of a model should help to guide efforts to collect additional data.

To illustrate the problem of lack of identifiability, imagine a situation where we have a parsimonious model that is a perfect representation of the real system and that there are no input or output errors on the measured data, then the model output that exactly reproduces the observed output will uniquely identify the correct parameter set. However, this situation is rarely encountered in environmental modelling, and in the presence of model and data error, different parameterisations can produce outputs with similar performance statistics and therefore it is not possible to say with certainty which model is correct. This effect is termed *equifinality* by Beven and Binley (1992). Note that the term model here refers to a set of model parameters within a fixed model structure, although similarly, different model structures may yield equivalent performance. Although many of the techniques contained in the toolbox can be extended to the analysis of multiple model structures, this aspect is not addressed in this version.

The classical method of addressing this problem is to turn to the use of system identification techniques where statistical or stochastic models are produced with parameters estimated as probability distributions. However, system identification techniques are often limited to linear systems and are subject to many statistical assumptions. An alternative approach that can be used with nonlinear models is Monte-Carlo simulation (e.g. Press *et al*., 1996). However, this technique requires multiple model runs to be performed and therefore is very computationally expensive. Assumptions of the nature of data errors are also required in the MC simulation approach (see also Appendix B, that describes a function to compare different approaches to uncertainty estimation using a simple example).

Monte-Carlo parameter sampling provides an alternative approach to classical statistical modelling and uncertainty estimation methods and this toolbox is designed to help interpret the results of such modelling experiments.

GLOSSARY

Objective Function	A criterion of model performance, e.g. sum of squared errors. Low values (within MCAT) indicate improved performance (fit).
Behavioural	Models that give sufficiently (to be defined by the modeller) good results in terms of one or more objective functions that the model is considered to be consistent with the observed response of the system under investigation.
Likelihood	Value that indicates the likelihood that a model (defined as a parameter set) is a good descriptor of the system. Higher values indicate a better representation of the system response.
Pareto Set	*'A solution is said to be Pareto optimal* [i.e. part of the Pareto set] *… if the value of any objective function cannot be improved without degrading at least one of the other objective functions*' (Chankong and Haimes, 1993).
Model	A combination of a model structure and a parameter set (within this structure).

PUBLICATIONS

Wagener T., Lees M.J., Wheater H.S. 1999. A generic rainfall-runoff modelling toolbox. *Eos Trans. AGU*, 80, Fall Meet. Suppl., F203.

Wagener, T., Lees, M.J., Wheater, H.S. 2000. Incorporating predictive uncertainty into a rainfall-runoff modelling system. Proceedings of Hydroinformatics 2000 Conference, on CD, Iowa, USA.

Wagener T., Boyle D.P., Lees M.J., Wheater H.S., Gupta H.V., Sorooshian S. 2001. A framework for the development and application of hydrological models. *Hydrology Earth System Sciences*, **5(1)**, 13-26.

Wagener T., Lees M.J., Wheater H.S. 2001. A framework for the development and application of parsimonious hydrological models. To appear in Singh, Frevert, Meyer (eds.) Mathematical models of small watershed hydrology – Volume 2. Wat. Resour. Publ. LLC, USA.

Wagener, T., McIntyre, N., Lees, M.J., Wheater, H.S., Gupta, H.V. 2001. Towards reduced uncertainty in conceptual rainfall-runoff modelling: Dynamic identifiability analysis. *Hydrological Processes*, in Press.

Wagener T., Camacho L.A., Lees M.J., Wheater H.S. 2001. Dynamic parameter identifiability of a solute transport model. *Proceedings of EuropIA'8*, Delft, April 2001, NL.

Sincock, A.M., Wheater, H.S., Whitehead, P.G.. Calibration, sensitivity and uncertainty analysis of a river water quality model under unsteady flow conditions. *Journal of Hydrology*, in Review.

2

MONTE-CARLO PARAMETER SAMPLING

Monte-Carlo parameter sampling refers to repetitive model simulation with parameters randomly sampled from probability distributions (e.g. uniform or normal). Usually the model is run in a batch mode with a randomly generated matrix of parameter values as inputs, and model output and objective function values stored for each model run in an output matrix. Often, no prior information regarding the nature of the parameter distributions and their correlation is available, so uniform distributions are used. This is sometimes referred to as a diffuse prior. If uniform distributions are used, only the boundaries must be specified. The boundaries must be set wide enough such that any possible behavioural parameter combination is simulated. However, if the parameter boundaries are set too wide then sparse sampling of the parameter space ensues. The number of simulations determines how extensively the parameter space is sampled, and as large a number of simulations as is computationally possible (a function of model run-time and memory) should be performed. However, other termination criteria are possible, e.g. stability of mean etc. If computation time is an important constraint then techniques such as Latin-hypercube or stratified sampling can be used to increase sampling efficiency. However, these approaches are not without limitations which have to be considered (see for example Press *et al*., 1996).

Population evolution variants of the Monte-Carlo method involve the use of an optimisation procedure to evolve the randomly sampled parameter space. This can be thought of as the introduction of a behavioural objective in order to achieve denser sampling in the parameter region(s) of interest (*i.e.* one or more where better performing parameter sets are located). The University of Arizona research group have produce a shuffled complex evolution algorithm that combines random parameter sampling with a simplex optimiser and a shuffling procedure (SCE-UA, see Duan *et al*., 1992) and have extended this technique to the multi-objective case (MOCOM-UA, see Yapo *et al.*, 1998). Montesinos and Beven (1999) used a genetic algorithm (Goldberg, 1989) to sample the feasible parameter space. With this technique a large number of behavioural models (parameter sets) can be produced with less computational expense than would be the case using pure Monte-Carlo sampling.

3

PARAMETER AND OUTPUT UNCERTAINTY

The core of the toolbox is based on the concept of Regional Sensitivity Analysis (RSA, Spear and Hornberger, 1980; Hornberger and Spear, 1981), and its extension to the Generalised Likelihood Uncertainty Estimation (GLUE) technique developed at Lancaster University by Beven and co-workers (Beven and Binley, 1992; Freer *et al*., 1996). In GLUE no single optimum parameter set is identified, rather a set of behavioural models are selected where each model has a certain likelihood (pseudo probability) of being the correct representation of the system. This method recognises that it is not possible to identify a single best model in the face of model/data error. Lancaster University has produced a GLUE software tool that has some common functionality with MCAT, see,

http://www.es.lancs.ac.uk/es/research/hfdg/topmodel.html#GLUE

The GLUE methodology introduces the concept of likelihood in a wider context than the traditional statistical one. Likelihoods are any performance measures that can be used to differentiate how likely it is that the model (*i.e.* a specific parameter set and model structure combination) is representative for the system at hand. Likelihoods must be monotonically increasing, above zero and add up to one. MCAT transforms objectives into likelihoods according to the following pseudo code,

LHOOD = 1-OBJ

IF MIN (LHOOD) < 0 THEN LHOOD = LHOOD – MIN (LHOOD)

LHOOD = LHOOD / SUM(LHOOD)

Where *OBJ* is the objective function value and *LHOOD* is the resulting likelihood value.

Since GLUE is a likelihood-based (pseudo-probability) technique, it can be set within a Bayesian framework with posterior parameter and output uncertainties calculated on receipt of new information.

A major practical problem in application of the GLUE methodology comes with the subjective choice of likelihood function and the assessment of a likelihood threshold between behavioural and non-behavioural models, *i.e.* how bad does the performance of a model have to be before it can be rejected as having no probability of representing the system? Although this process appears rather arbitrary it is in many ways similar to the arbitrary assumptions that are made as part of the statistical modelling procedure, such as the nature of the error model. The flexibility of the MC sampling approach enables users to tune the uncertainty estimation procedure through the selection of objective functions, and through expert decisions regarding the expected performance of a model in the face of data errors and lack of knowledge of the underlying physical processes. Performance of the uncertainty estimation method can, and should, be tested by verification on independent data, in a similar fashion to split-test model calibration and validation.

The simplest method of removing non-behavioural models from the model set resulting from parameter sampling is to specify a behavioural/non-behavioural threshold, *i.e*. all parameter sets producing an objective function value above this threshold are deleted. A different, multi-objective based, approach to determining which models are accepted as behavioural is proposed by researchers at the University of Arizona (Gupta *et al*., 1998; Sorooshian *et al*., 1998; Yapo *et al*., 1998; Boyle *et al*., 2001). Here, behavioural models are defined as models that describe certain characteristics of the response well. Formally the set of behavioural models is defined objectively using a Pareto definition (see Appendix A). The technique recognises that in the presence of model structural errors the selection of different optimisation objectives will result in different optimal parameter sets (Sefe and Boughton, 1982). It is therefore not possible to find a single optimum parameter set, but rather a set of solutions where no real distinction is possible, a so-called Pareto set of solutions. None of the members of the Pareto set produces the best result for all objective functions. The members of the Pareto set are also referred to as non-dominated parameter sets, *i.e.* they cannot be distinguished from each other as better representations of the system. Parameter sets are placed outside the Pareto set when at least one set inside is better with respect to all objective functions. Wagener *et al*. (2000) compare a number of different methods of selecting a behavioural model set additionally to Pareto ranking.

Once a behavioural model set has been identified using either the GLUE or Pareto methods, predictive uncertainty bounds can be produced for the output time-series. In the GLUE case percentile model output confidence intervals are produced whereas in the Pareto case the bounds represent the envelopes of predictions produced by the whole model set and by the Pareto set.

The issue of choice of likelihood function has been widely explored (see Beven and Binley, 1992; Freer *et al*., 1996, amongst others). The MCAT toolbox allows users to investigate the effect of different likelihoods on the parameter likelihood distribution and output uncertainty. Note that likelihoods can be constructed from multiple objectives, allowing the incorporation of additional information about the performance of a model to represent the real system, to be incorporated in the uncertainty estimation procedure.

4

MCAT FUNCTIONALITY

Table 1 provides an overview of the menu options available in MCAT. Only brief descriptions are given here, since more detailed explanations are given later.

Table 1 Brief description of MCAT menu options.

MCAT MENU OPTIONS	
MENU OPTION	FUNCTIONALITY
Parameter Table	Plots the top 10 parameter sets with respect to the currently selected objective function in the command window.
Dotty Plots (Objective Functions)	A projection of the response surface defined by the currently selected objective function into one dimension for every parameter value.
Dotty Plots (Likelihoods)	A projection of the response surface defined by the likelihood (based on the currently selected objective function) into one dimension for every parameter value.
A Posteriori Parameter Distribution	The parameter distribution conditioned on the likelihood values calculated from the currently selected objective function.
Identifiability Plot	Representation of parameter identifiability using the gradient of the cumulative likelihood distribution.
Regional Sensitivity Analysis	The parameter population is sorted according to the currently selected objective function and split into ten equally sized groups. The cumulative distributions for each group are plotted. A difference in the distributions suggests sensitivity of the model performance to the parameter analysed.
Parameter View	Two and three dimensional plots of the response surface with respect to the currently selected objective function.
Class Plot	The parameter population is sorted according to the currently selected objective function and split into ten groups of equal size. The best parameter set of each group is used to produce a hydrograph which is plotted together with the observed data.
GLUE Variable Uncertainty	Plot of the cumulative likelihood distribution and the likelihood distribution of the selected output variable.
GLUE Output Uncertainty	Prediction confidence limits plotted using the GLUE methodology.
PARETO Output Uncertainty	Model output time series are calculated using every parameter set in the Pareto population. The highest and lowest calculated values at every time step are

	used to define the Pareto uncertainty limits.
Multi-Objective Plots	All objective functions are plotted against each other in order to identify correlated measures.
Normalised Parameter Range Plot	The best (normalised) parameter values with respect to the different objective functions and the Pareto range (if calculated) are shown.
Best Predictions Plot	Plotting the best simulation with respect to the different objective functions.
Threshold	A threshold value can be used to narrow down the population of behavioural models.
Set Upper Cl	Select the upper GLUE uncertainty limit (default is 0.95).
Set Lower Cl	Select the lower GLUE uncertainty limit (default is 0.05).
Zoom In	Enlarges a part of the current plot.
Exit	Close MCAT and return to the Matlab command window.

5

DEMONSTRATION EXAMPLES

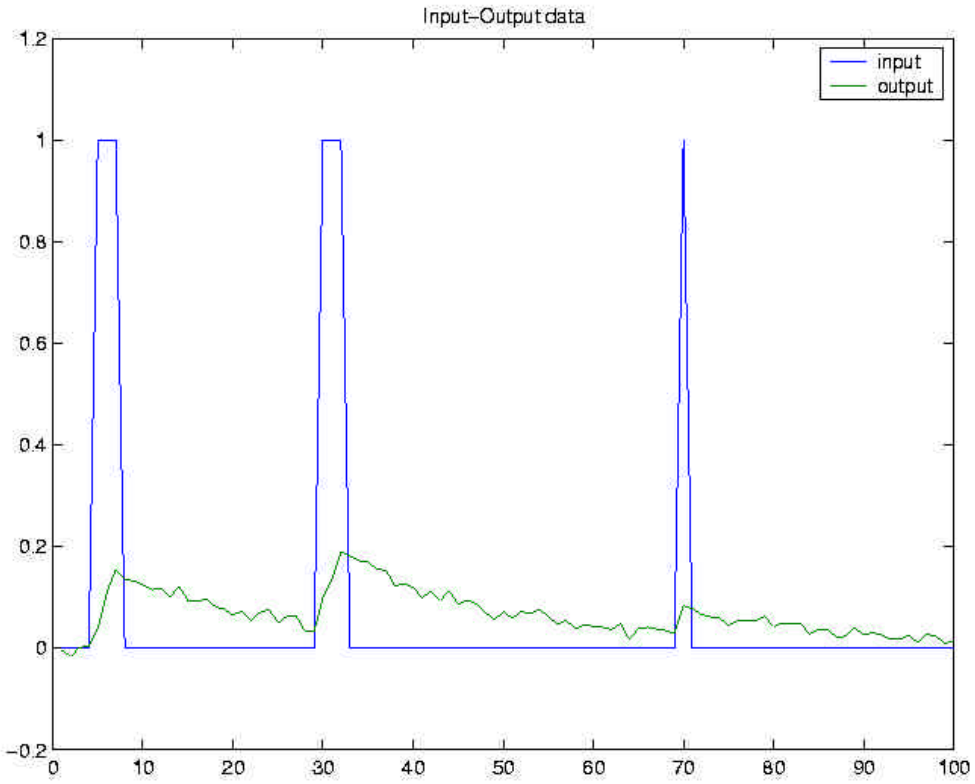


Figure 1 Synthetic input and output data from the first example.

5.1 EXAMPLE 1

Throughout this user manual simple modelling examples are used to illustrate the functionality of the toolbox. The following discretised linear storage model is used,

$$y(i)=(1-1/rt)*y(i-1)+(1/rt)*gain*u(i)$$

where y is the output at sample i, u is the input and the model parameters are the residence time (rt) and gain ($gain$).

A heavily commented sample batch file called ‘mcat_eg1.m’ is included in the MCAT directory. To run this example, type ‘mcat_eg1’ after installing the toolbox.

In this example, synthetic error-corrupted data are first generated with parameter values of rt=20 and $gain$=1. These can be considered to be the ‘true’ parameters. Next, 1000 Monte-Carlo simulations of this linear storage model are run with parameters sampled from uniform distributions with boundaries for the uniform distributions set at rt=10-30 and $gain$=0.5-1.5.

The input-output data are shown in Figure 1.

5.2 EXAMPLE 2

A second example file called ‘mcat_eg2.m’ is also included in the MCAT directory. This example uses a more complex model consisting of the combination of two linear stores in parallel, in contrast to the single storage described above. Both stores have different residence times and gains. The form of this example is largely similar to the first one, except that the data generated are error free and an objective function based on one component of the response (low output) is also produced to allow investigation of the benefits of multiple-objectives. It is interesting to compare this slightly more complex model (4 parameters) with the simpler model (2 parameters) analysed in the first example.

5.3 EXAMPLE - COMPARISON

This example (‘mcat_comp.m’) compares different methods of parameter/output uncertainty estimation. It uses the simplest dynamic model possible, an autoregressive model with a single parameter. The methods compared are:

1. Theoretical least squares regression (see for example Press et al., 1992).

2. Classical Monte Carlo simulation (see for example Press et al., 1992).

3. Generalised Likelihood Uncertainty Estimation (GLUE, Beven and Binley, 1992) using MCAT.

More information on this example is provided when going through the comparison within Matlab. Figures of the main example results can be found in Appendix B.

6

USING MCAT

6.1 INSTALLING MCAT

Copy the MCAT folder to your local toolbox directory (usually C:\MATLAB\toolbox\local). Add the MCAT directory to the MATLAB path using the path browser ().

MCAT has been developed in MATLAB 5.2[1] (Release 11) and as such it is not guaranteed that it will work in earlier releases. The MATLAB functions (with the exception of the examples) are provided as p-code and as such cannot modified. If you find any problems with the toolbox, please report it to us with a copy of the input data saved as a '.mat' data-file. **We also welcome any suggestions for improvements.**

A number of operations is computationally intensive, such as computation of the Pareto set, while other functions such as the parameter surface plotting are memory intensive. If operations take a very long time or you run out of memory try reducing the number of Monte-Carlo simulations and/or the length of the time-series entered into the program.

6.2 STARTING MCAT

The results from your Monte-Carlo sampling experiment need to be formulated into a specified format prior to input to the MCAT program. If the Monte-Carlo simulation has been carried out outside the MATLAB environment then the results must be loaded into MATLAB first. Type, `>> help mcat` for a description of the input variables and format required,

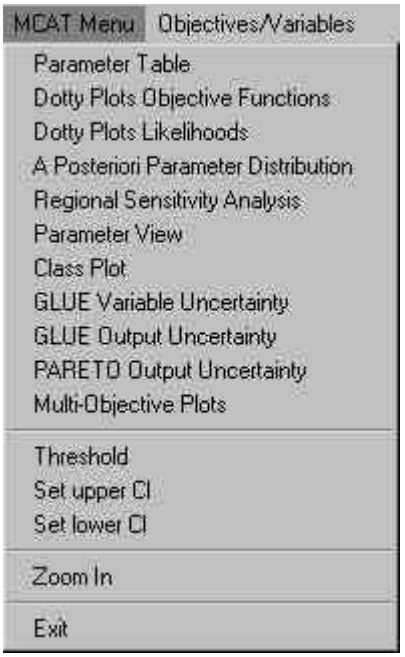
```
function mcat(pars,crit,vars,mct,pao,obs,id,pstr,cstr,vstr,dt,t);

Interface between monte-carlo simulation output and MCAT

  pars: MC parameter matrix [no. of sims x no. of pars]
  crit: MC objective matrix [no. of sims x no. of objective]
  vars: MC variable matrix [no. of sims x no. of pars]
  mct:  MC output time-series matrix [no. of sims x no. of samples]
  pao:  Pareto population matrix [no. of sims x no. of pars]
  obs:  observed time-series vector [no. of samples x 1]
  id:   descriptor [string]
  pstr: parameter names [string matrix - use str2mat to create]
  cstr: parameter names [string matrix - use str2mat to create]
  vstr: parameter names [string matrix - use str2mat to create]
  dt:   sampling interval in minutes
  t:    time vector if irregularly spaced samples

  Enter an empty matrix '[]' for null inputs
```

[1] A separate collection of p-files is available for use in Matlab 6.0 (Release 12).

6.3 MCAT FUNCTIONS

MCAT is started by typing mcat (pars, crit, vars, mct, pao, obs, id, pstr, cstr, vstr, dt, t);. Assuming that the inputs are in the correct format, the MCAT graphical user interface will open. This GUI has two main menus.

The MCAT MENU shown below provides access to a number of different analysis and visualisation tools that are described below. Note that some menu options will not appear if the input data required are not available.



A second menu allows the user to select the objective or variable to be used in the analysis. This menu is different according to the objectives and variables generated from the Monte-Carlo simulation. The example shown below is for the linear storage example: here, *sse* is the sum of the squared errors between the model output and the actual output (synthetic in this case); *abias* is the absolute bias (*i.e.* sum of the errors); and peak output is the maximum model output.

6.3.1 PARAMETER TABLE

The parameter table option prints the top ten parameter sets and their corresponding objective function value(s) in the command window. The sorting of the parameter sets is based on the currently selected objective function as displayed in the header of the printed table.

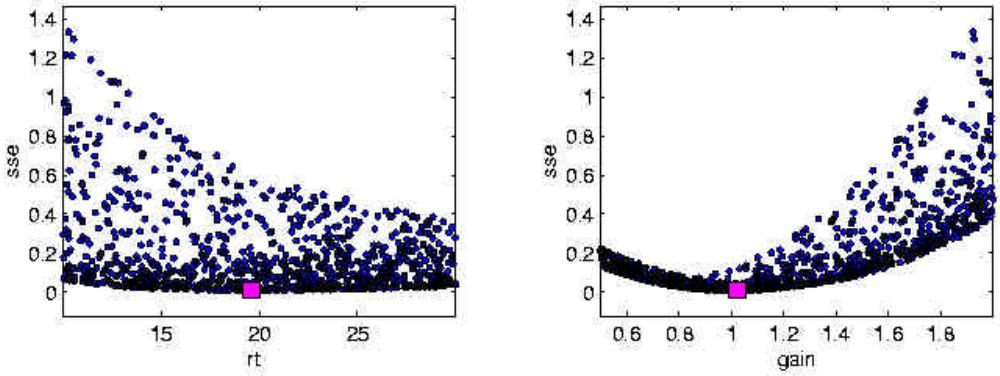


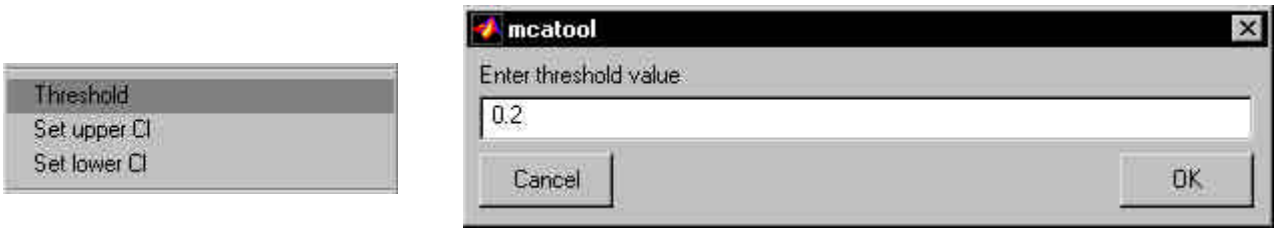
Figure 2 Dotty plots objective function (*sse*).

6.3.2 DOTTY PLOTS (OBJECTIVE FUNCTIONS)

Dotty plots represent a projection of the parameter space into 1 dimension. Each dot represents the objective or variable value associated with a single parameter set. If the 'surface' of the dotty plot (representing the best model parameterisation for a given value of the parameter) has a clearly defined minimum then the parameter can be considered to be well identified.

Example dotty plots for the linear storage model example are shown in Figure 2. Note that although the parameter set that has the lowest *sse* is close to the 'true' value there are many other parameter sets (especially values of *rt*) that give *sse* values almost as good as the best value.
The best parameter value is shown as a magenta rectangle on the dotty plot.

The dotty plot shows that there is a considerable range of model performance produced by the sampled model parameter sets. Clearly some of these models have produced output than can be considered to be non-behavioural, *i.e.* the response deviates so far from the observations that the model cannot be considered to be a possible characterisation of the system. These non-behavioural models can be removed from the analysis by setting a threshold from the MCAT menu.



The dotty plots in Figure 3 show the result of setting a rejection threshold of *sse*=0.2. Clearly the *gain* parameter appears to be better defined within this set of behavioural models.
The cyan diamonds show the Pareto set which is defined later in the text.

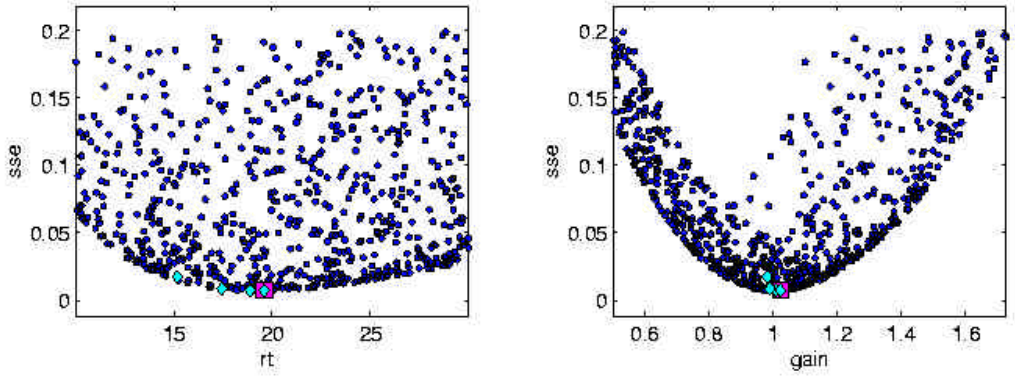


Figure 3 Dotty plots objective function (*sse*): threshold = 0.2.

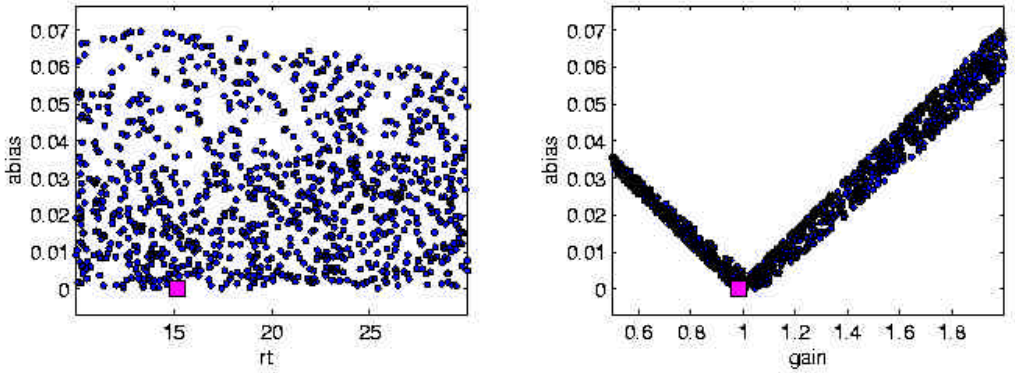


Figure 4 Dotty plots objective function (*abias*).

The dotty plot shown in Figure 4 is for the absolute bias criterion. It can be seen that the residence time parameter is unidentifiable in this case, whereas the gain parameter is clearly identifiable. This result is unsurprising since the gain parameter controls the relationship between the input and output totals. This plot also shows the Pareto population calculated when selecting the Pareto output uncertainty option from the menu.

6.3.3 DOTTY PLOTS (LIKELIHOODS)

This plot uses the likelihoods (calculation as described earlier) instead of the objective function values to produce dotty plots (Figure 5). The best parameter value is now the point with the highest likelihood value, in contrast to the dotty plot using the objective function where the lowest dot indicates the best value.

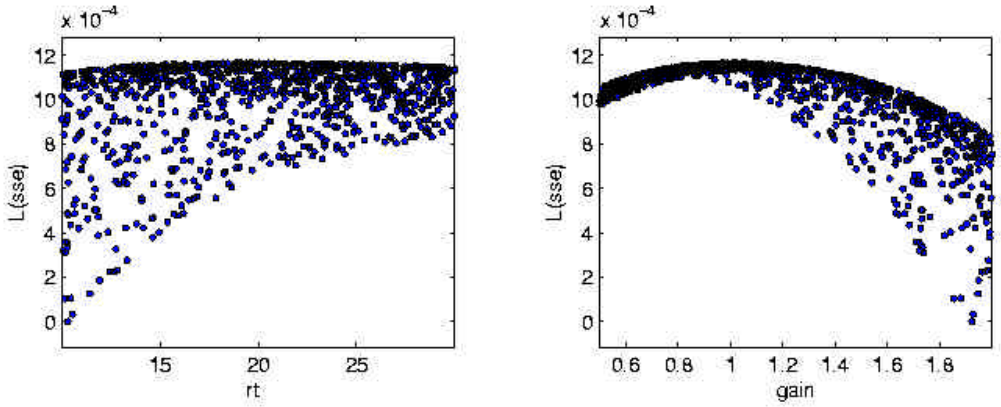


Figure 5 Dotty plots likelihood (based on sse objective function).

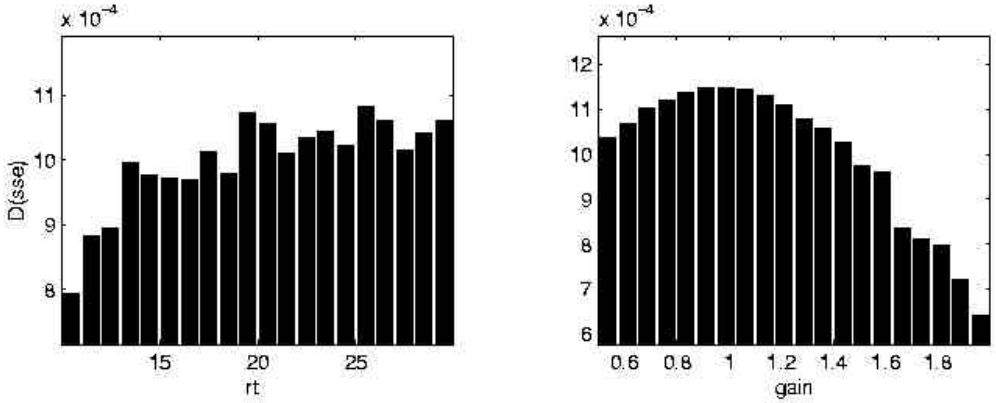


Figure 6 A *posteriori* parameter distribution (based on sse objective function).

6.3.4 A POSTERIORI PARAMETER DISTRIBUTION

In this plot, the range of every parameter is divided into 20 containers of equal width. The likelihoods in each container are added up and divided by the sum of parameter values within the container. This gives the distribution of the parameter after being conditioned using the selected objective function. The resulting distribution values *D* are plotted as bars as shown in Figure 6 (remember that the initial population was uniformly distributed).

6.3.5 IDENTIFIABILITY PLOT

This plot uses the gradient of the cumulative likelihood distribution as a measure of identifiability of the parameters (Figure 7). The best performing 10% of each parameter population are selected and the cumulative distribution is calculated. The parameter ranges are split into 10 bins and the gradient is calculated in each. A higher gradient is additionally indicated by a darker grey shading.

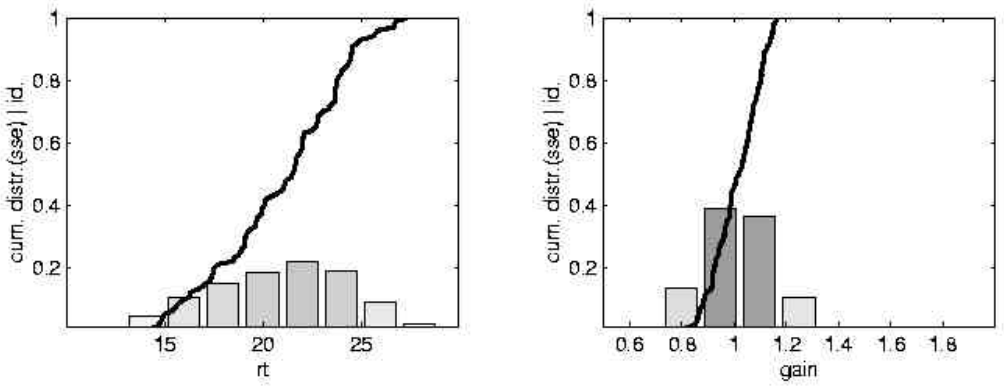


Figure 7 Identifiability plot (sse).

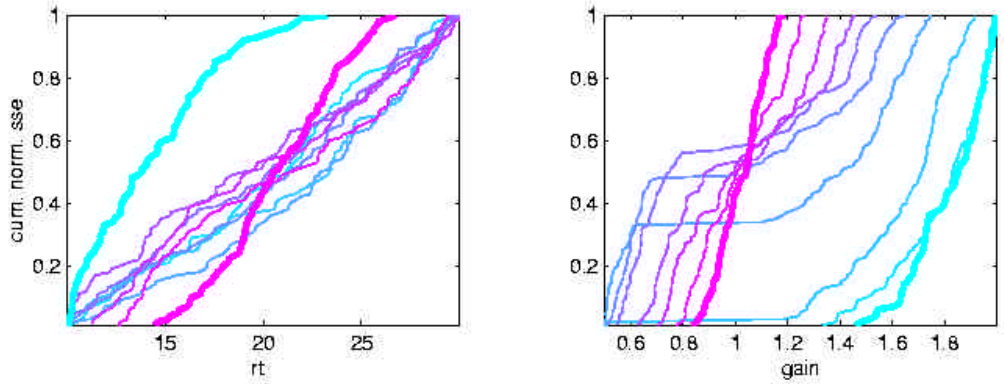


Figure 8 Regional sensitivity plot (*sse*).

This plot is sensitive to the number of parameter sets selected. The selection has to be representative for the top of the population, *i.e.* the response surface.

6.3.6 REGIONAL SENSITIVITY ANALYSIS

Regional Sensitivity Analysis (RSA, Hornberger and Spear, 1981; Spear and Hornberger, 1980) is a method of assessing the sensitivity of model parameters where sensitivity is defined as the effect of the parameters on overall model performance (as indicated by objectives) or specific output variables such as peak output, total output, etc.

The approach adopted here uses the extension introduced by Freer et al. (1996). The parameter sets are ranked according to the selected objective, which is transformed to a likelihood value as explain earlier. The parameter sets are then split into 10 groups. For each group the likelihoods are normalised by dividing by their total, and the cumulative frequency distribution is calculated and plotted. If the model performance is sensitive to a particular parameter there will be a large difference between the cumulative frequency

distributions; *i.e.* the parameter has a significant effect on the model output. Sensitivity analysis can also be carried out with respect to variables by selecting the desired variable from the 'objectives/variables' menu.

The sensitivity plots for the linear storage example shown in Figure 8 reveal that both parameters are sensitive and that the gain parameter is more sensitive than the residence time parameter.

6.3.7 PARAMETER VIEW

(only available if there are 2 or more parameters)

This tool allows the visualisation of 2-D parameter surfaces, *i.e.* interpolated plots of objective values (z-axis) for 2 selected parameters (x/y-axis). In order to examine parameter interaction a Monte-Carlo simulation should be run with just 2 parameters varied and the other parameters fixed, otherwise the surface will be noisy since parameter pairs in close proximity will have large differences in the objective or variable value caused by variations in other parameters (cf. scatter in dotty plots). The Pareto set is also displayed if the contour plot option is selected.

Selecting the parameter view menu option opens a new window with a number of menus. These menus can be used to select the parameters, objectives/variables, plot type (contour, 2-D surface, 3D surface, mesh), colour map used, and viewing angle.

The plot in Figure 9 shows the parameter surface for the linear storage example. Note that there is a relatively well-defined region of low *sse*, which is smaller for the gain parameter than for the residence time parameter. The Pareto set are indicated as blue diamonds. However, random sampling is not an efficient method to derive the Pareto set. Algorithms based on population evolution methods can be used for that purpose (Yapo *et al*., 1996; Khu, 1998).

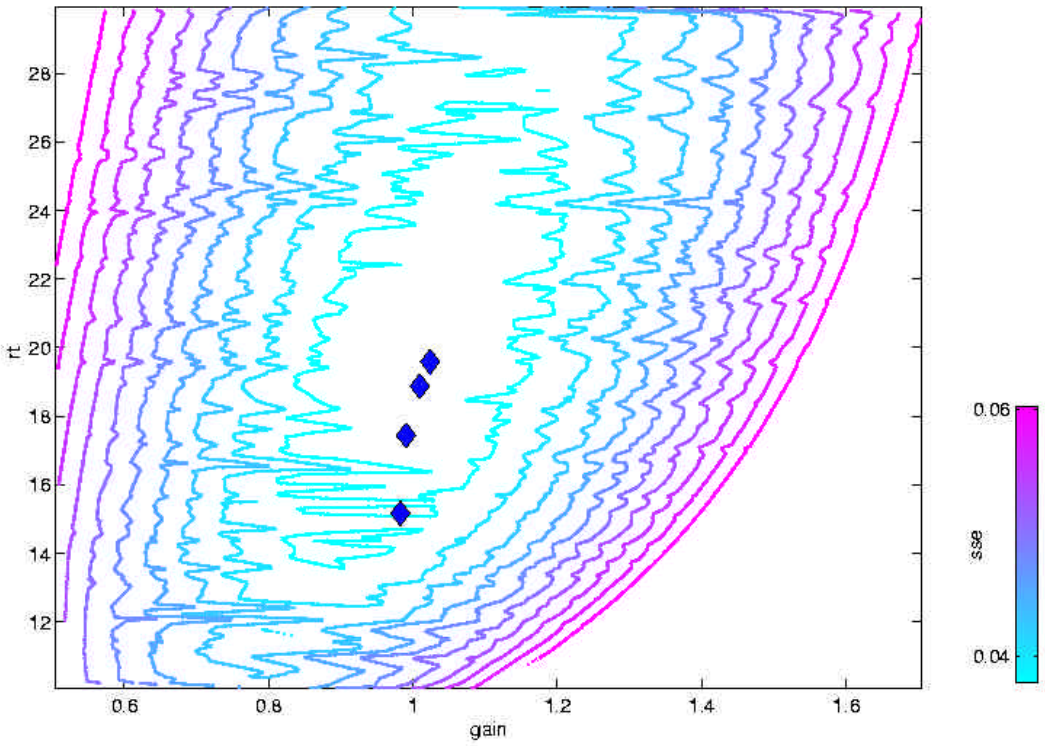


Figure 9 Contour plot of *gain* vs. *rt* for the linear storage model example.

6.3.8 CLASS PLOT

(only available if Monte-Carlo time-series are input to MCAT)

The class plot displays 10 sample Monte-Carlo simulation time-series outputs, classified according to the objective function or variable specified. If an objective is specified it is transformed to a likelihood. This plot gives a useful indication of the spread of output time-series. For example, the plot in Figure 10 shows the best and worse simulations (as measured by *sse*) for the linear storage example as well as 9 intermediate simulations. It is possible to classify parameter sets according to output response to reduce the computational requirement in prediction situations (see for example, Franks and Beven, 1997).

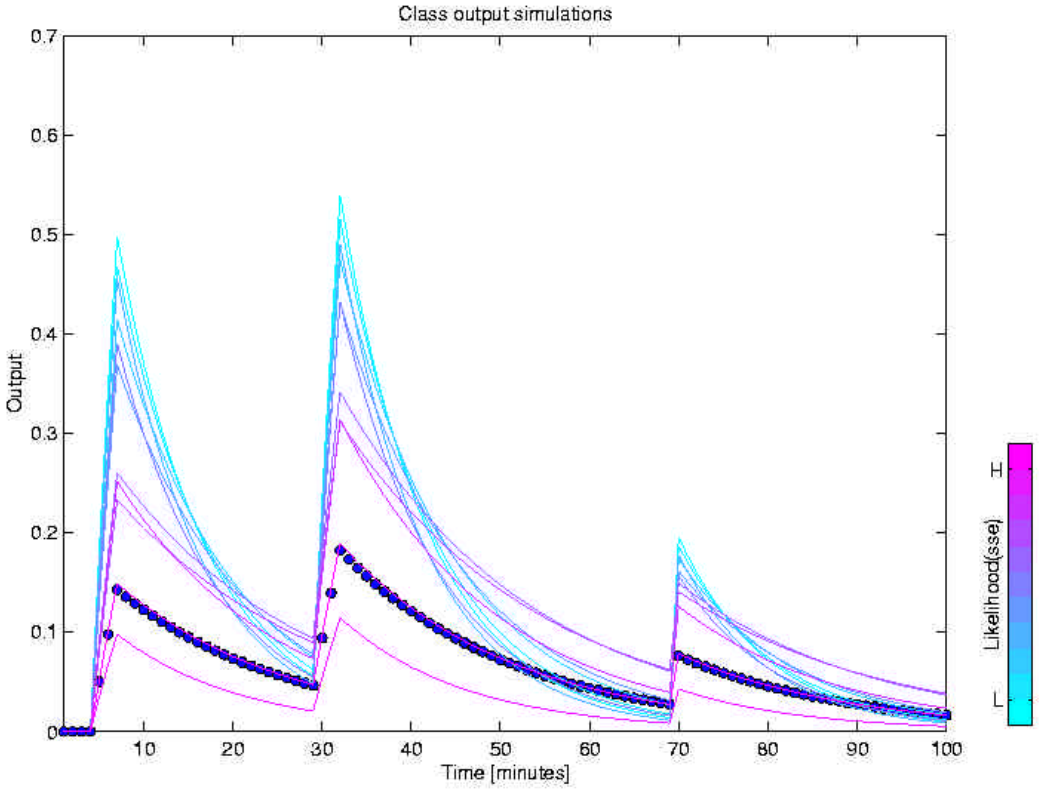
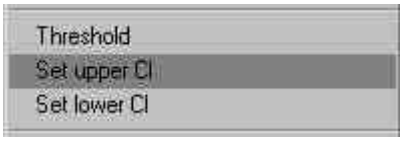


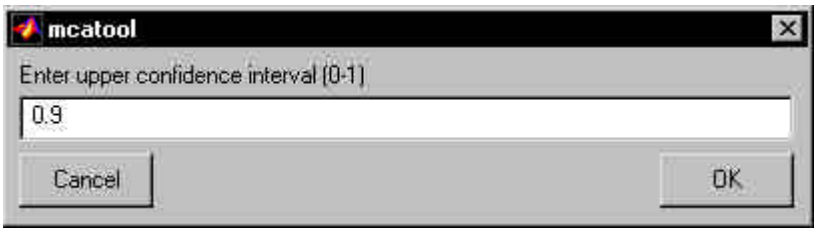
Figure 10 Class plot.

6.3.9 VARIABLE UNCERTAINTY

(only available if variables are input to MCAT)

The variable uncertainty plot shows the cumulative probability distribution and probability density function for a selected variable calculated using a selected objective (transformed to likelihood). Confidence intervals can be set from the MCAT menu.





The plot in Figure 11 shows the uncertainty in peak output for the linear storage example. The result show that there is a probability of 0.05 (5%) that the peak output will be above 0.50 or below 0.10.

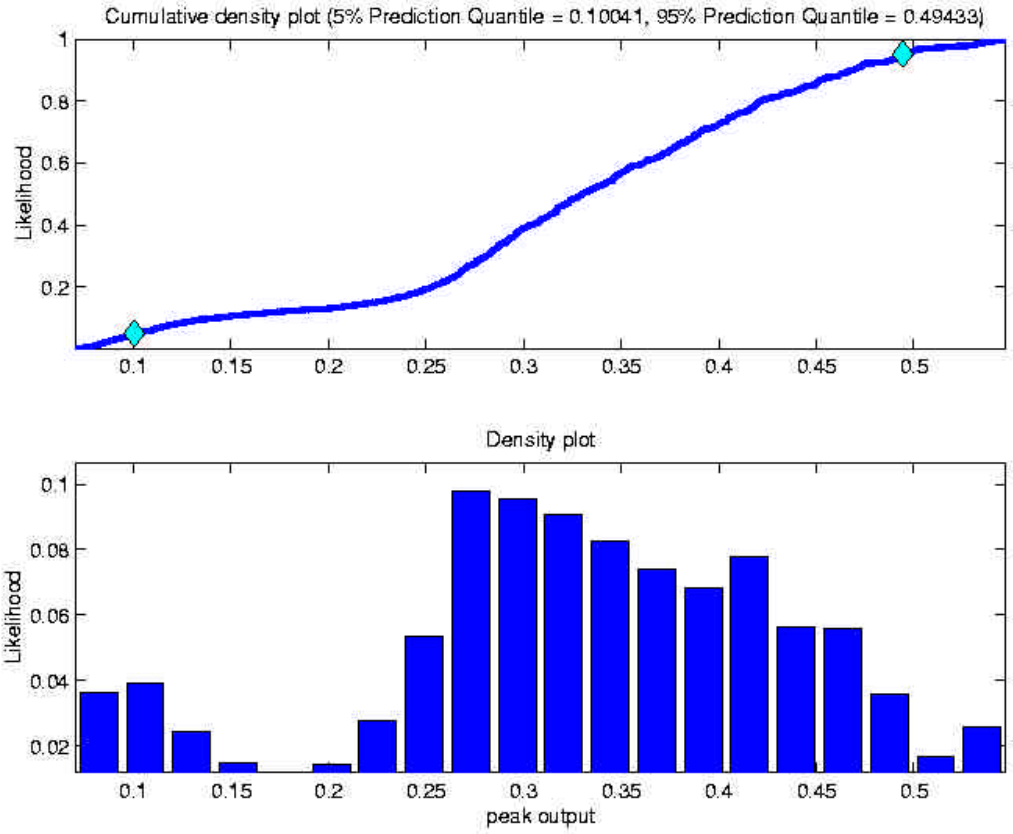


Figure 11 GLUE variable uncertainty plot.

6.3.10 OUTPUT UNCERTAINTY

(only available if Monte-Carlo time-series are input to MCAT)

This plot displays the time-series output with associated confidence intervals calculated using the GLUE methodology. For each point in time a cumulative frequency distribution is generated using the selected objective (converted to a likelihood) and the confidence intervals calculated using linear interpolation. A measure of the width of the confidence intervals normalised by the mean is also given below. The confidence intervals can be specified from the MCAT menu as described in the section on variable uncertainty.

The plot in Figure 12 shows the linear storage model output with associated 95% confidence intervals. The bottom graph is added to make the identification of regions with large uncertainties easier. It shows the difference between upper and lower confidence limit, normalised by the maximum difference between the two over the investigated period of time, or in the form of an equation (remember that this is a relative measure and that it is of no use when comparing two different plots):

$$\Delta CFL_i = \frac{UCFL_i - LCFL_i}{\max(UCFL - LCFL)}$$

where

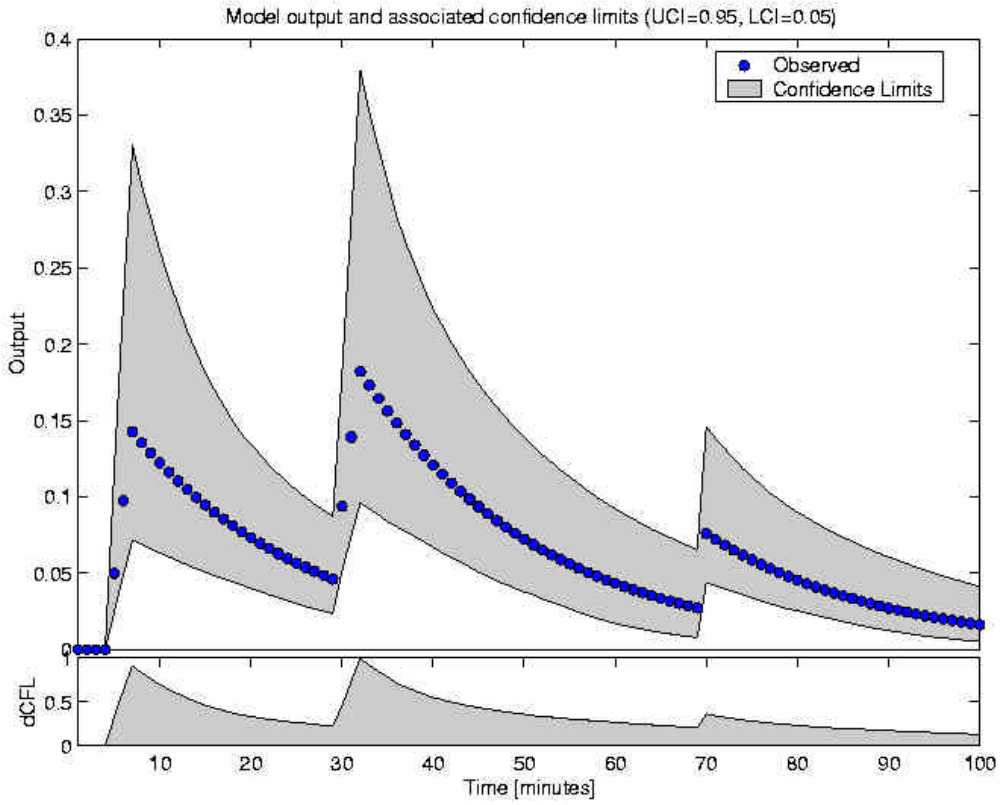


Figure 12 GLUE output uncertainty limits.

ΔCFL	normalised difference between upper and lower confidence limit
UCFL	upper confidence limit
LCFL	lower confidence limit
max( )	maximum difference over time series

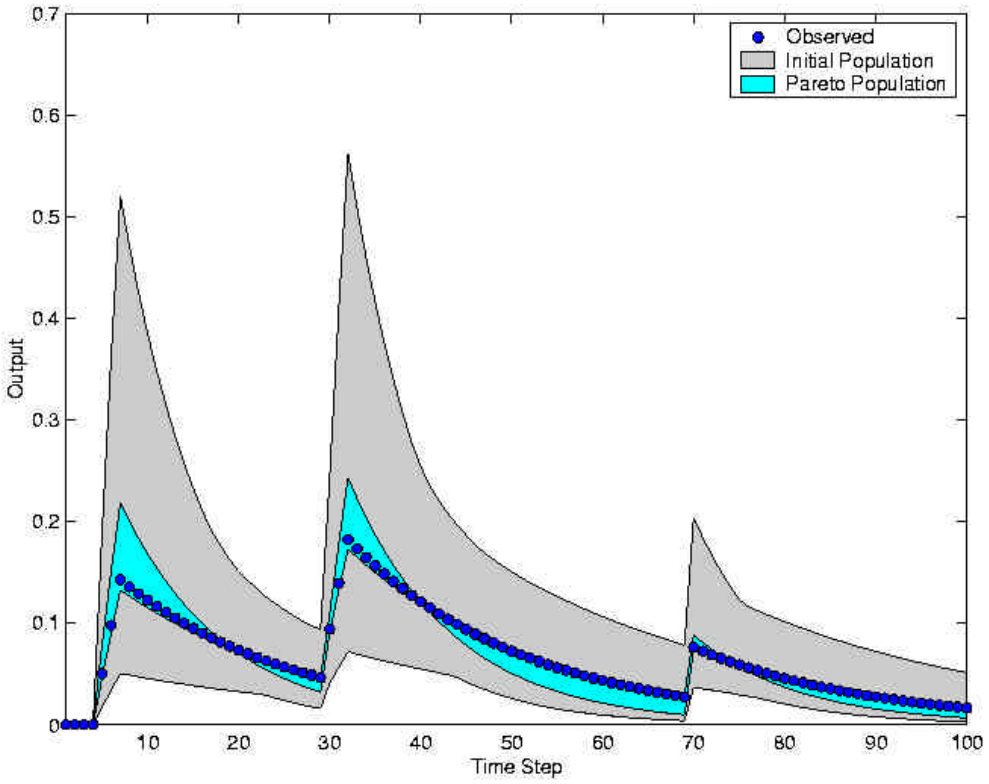


Figure 13 Pareto output uncertainty limits.

6.3.11 PARETO OUTPUT UNCERTAINTY

(only available if Monte-Carlo time-series are input to MCAT)

The Pareto output uncertainty plot (Figure 13) shows the range of model outputs produced by parameter sets from the *a priori* parameter space (specified by the Monte-Carlo boundaries) in grey (outer bound) and the output response of models in the Pareto set, *i.e*. parameter sets with rank one (see Appendix A).

6.3.12 MULTI-OBJECTIVE PLOTS

These are scatter plots of each objective versus every other objective function (Figure 14). A large scatter indicates that the objectives are unrelated, whereas a clear pattern indicates that the objective functions attempt to fit the model to the same part of the observed data, *i.e*. they retrieve similar information from the data set. This plot can be used to remove redundant objective functions. Points situated in the bottom left hand corner represent good models whereas those in the top right hand corner represent poor models. In many situations the best model as measured by one objective is not the best model according to a different criteria (in cases when the criteria are not strongly correlated). A trade-off front between the two objectives can be seen in those cases (see Appendix A). If the Pareto set has been

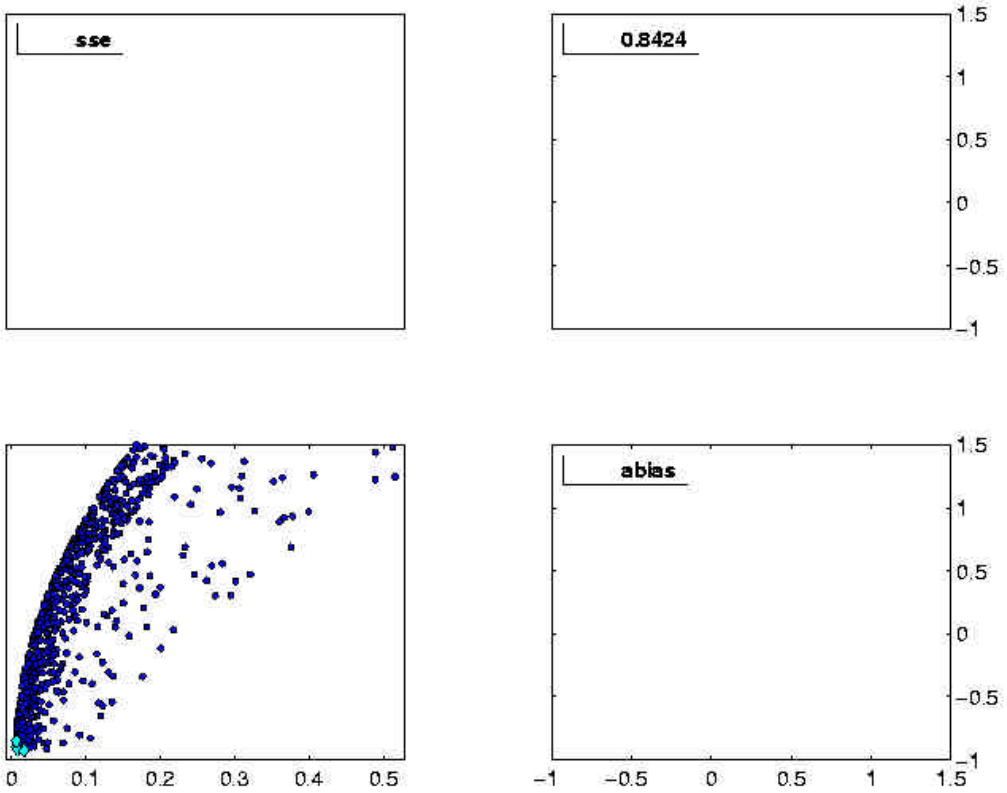


Figure 14 Multi-objective plot. The bottom left plot was zoomed in. The cyan diamonds are the Pareto set. The parameter ranges are not identical between the plots because the zoom was used.

calculated (by selecting the Pareto output uncertainty option) it is also displayed on the plot as cyan diamonds.

The objective function names are shown in the diagonal plots. The plot(s) 'above' the diagonal shows the correlation coefficients calculated in Matlab using the 'corrcoef' command.

6.3.13 NORMALISED PARAMETER RANGE PLOT

The ranges for all parameters are normalised and the best solutions for the different (if available) objective functions are plot in different colours (Figure 15). The Pareto solution are plotted in light grey if they are already calculated. If not, select the 'Pareto Output Uncertainty' first to calculate them.

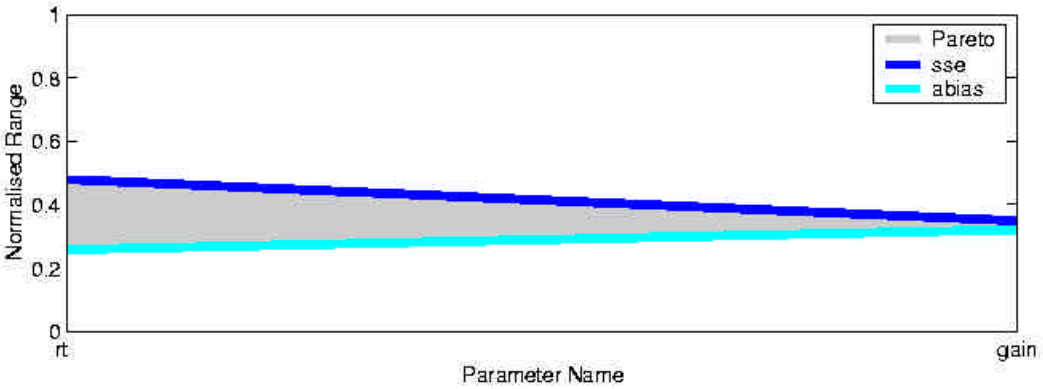


Figure 15 Normalised parameter range plot.

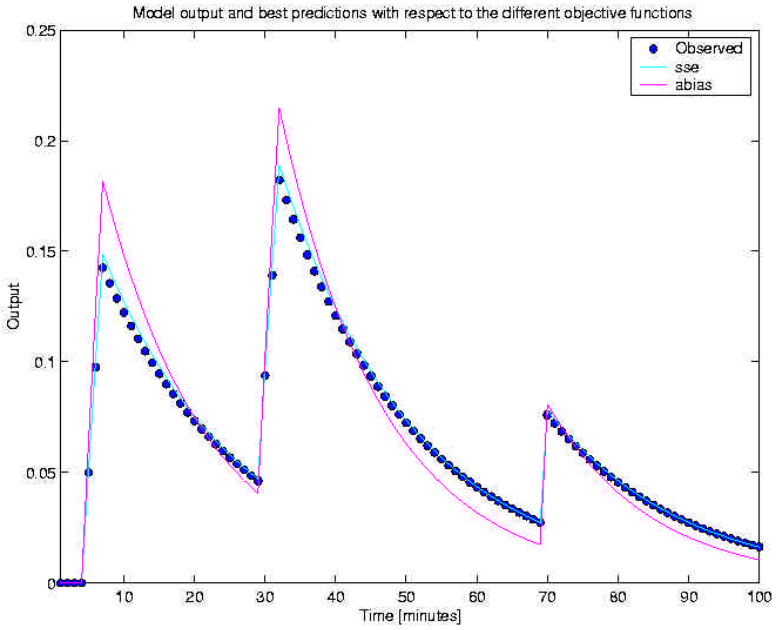


Figure 16 Best predictions plot.

6.3.14 BEST PREDICTIONS PLOT

(only available if Monte-Carlo time-series are input to MCAT)

This plot shows the best predictions with respect to the different objective functions available (Figure 16). The maximum number of predictions that can currently be displayed is seven.

6.3.15 ZOOM TOOL

Select zoom in from the MCAT menu and then select the area required for enlargement by dick and holding the left mouse button to defined a bounding box. To zoom out click the right mouse button on the figure (it works the other way round when you use the mouse with your left hand). The plot below shows an enlarged area of a multi-objective plot using the second example

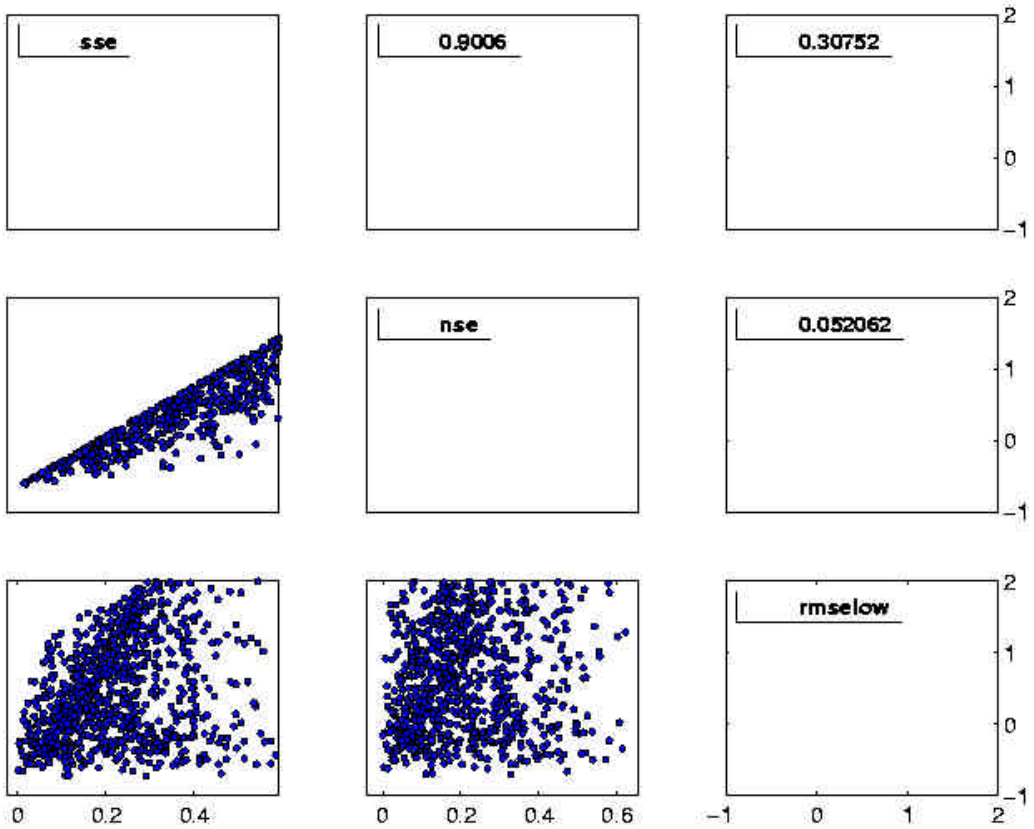


Figure 17 The sse measure versus nse (left plot), and the sse versus the rmse for the low flow period (right plot).

(mcat_eg2). The left plot shows the sum of squared error measure (sse) versus the Nash-Sutcliffe efficiency (nse, Nash and Sutcliffe, 1970) measure.

A high degree of correlation is visible. The right plot shows the sse measure versus the Root Mean Square Error (rmse) for the output period below a threshold value (0.05). Less correlation is visible indicating that both measures retrieve different types of information from the residuals.

This option is not required in Matlab 6 (Release 13) or higher since the graphic options are improved.

6.4 HOW TO ACCESS THE ACTUAL DATA

It might be necessary to access some of the actual data stored or even calculated within MCAT, e.g. the parameter sets with Pareto rank 1. This can be done using the following commands:

(1) Type: data = get(0,'userdata')
(2) Then type: data.'fieldname'
[e.g. data.pareto in order to access the Pareto set]

REFERENCES

Beven, K.J., Binley, A.M. 1992. The future of distributed models: model calibration and uncertainty prediction. *Hydrological Processes*, **6**, 279-298.

Boyle, D.P., Gupta, H.V., Sorooshian, S. 2000. Towards improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resources Research*, **36**, 3663-3674.

Chankong, V., Haimes, Y.Y. 1993. Multi-objective optimization: Pareto optimality. In Young, P.C. (ed.) Concise encyclopaedia of environmental systems. Pergamon Press, UK, 387-396.

Duan, Q., Gupta, V.K., Sorooshian, S. 1992. A shuffled complex evolution approach for effective and efficient global minimization. *Journal of Optimization Theory and Applications.* **76(3)**, 501-521.

Franchini, M., Pacciani, M. 1991. Comparative analysis of several conceptual rainfall-runoff models. *Journal of Hydrology*, **122**, 161-219.

Franks, S, Gineste, P, Beven, K.J., Merot, P. 1998. On constraining the predictions of a distributed model: the incorporation of fuzzy estimates of saturated areas into the calibration process. *Water Resources Research*, **34**, 787-797.

Franks, S., Beven, K.J. 1997. Estimation of evapotranspiration at the landscape scale: a fuzzy disaggregation approach. *Water Resources Research*, **33**, 2929-2938.

Freer, J., Beven, K.J., Ambroise, B. 1996. Bayesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach. *Water Resources Research*, **32(7)**, 2161-2173.

Goldberg, D.E. 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company, Inc., USA.

Gupta, H.V., Sorooshian S., Yapo, P.O. 1998. Toward improved calibration of hydrologic models: Multiple and non-commensurable measures of information. *Water Resources Research*, **34**, 751-763.

Hornberger G.M., Spear, R.C. 1981. An approach to the preliminary analysis of environmental systems, *Journal of Environmental Management*, **12**, 7-18.

Johnston, P.R., Pilgrim, D.H. 1976. Parameter optimization for watershed models. *Water Resources Research*, **12(3)**, 477-486.

Khu, S.-T. 1998. Automatic Calibration of NAM Model with Multi-Objectives Consideration. National University of Singapore / Danish Hydraulic Institute, D2K Technical Report 1298-1, DK.

Mein, R.G., Brown, B.M. 1978. Sensitivity of optimized parameters in watershed models. *Water Resources Research*, **14(2)**, 299-303.

Montesinos, P., Beven, K.J. 1999. Application of a genetic algorithm as sampler selector within the GLUE approach. Poster Presentation – European Geophysical Society Assembly, The Hague, NL.

Nash, J.E., Sutcliffe, J.V. 1970. River flow forecasting through conceptual models part I - A discussion of principles. *Journal of Hydrology*, **10**, 282-290.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flamery, B.P. 1992. Numerical Recipes in C: The art of scientific computing. 2nd Edition, Cambridge University Press, UK. (see also http://www.nr.com)

Sefe, F.T., Boughton, W.C. 1982. Variation of model parameter values and sensitivity with type of objective function. *Journal of Hydrology*, **21**, 117-132.

Sorooshian, S., Gupta, H.V., Bastidas, L. 1998. Final report - Calibration of hydrologic models using multi-objectives and visualisation techniques. Department of Hydrology and Water Resources, University of Arizona, USA.

Spear, R.C., Hornberger, G.M. 1980. Eutrophication in Peel Inlet, II, Identification of critical uncertainties via generalized sensitivity analysis. *Water Resources Research*, **14**, 43-49.

Wagener, T., Lees, M.J., Wheater, H.S. 2000. Incorporating predictive uncertainty into a rainfall-runoff modelling system. Proceedings of Hydroinformatics 2000 Conference, on CD, Iowa, USA.

Yapo, P.O., Gupta, H.V., Sorooshian, S. 1998. Multi-objective global optimization for hydrologic models. *Journal of Hydrology*, **204**, 83-97.

APPENDIX A

PARETO SETS

A range of objective functions are optimised simultaneously during multi-objective optimisation. These objective functions are usually noncommensurable and conflicting (Chankong and Haimes, 1993). In most cases, no solution will be available that is the best for all objective functions. Therefore only a solution population can be identified, which is a part of the feasible parameter population. A methodology to find this solution population was introduced by the economist Pareto at the end of the eighteenth century (Pareto, 1896). Chankong and Haimes (1993) define the concept as follows: *A solution is said to be **Pareto optimal** (also synonymously known in the literature as efficient, noninferior and **nondominated**) if the value of any objective function cannot be improved without degrading at least one of the other objective functions*.

PARETO OPTIMALITY

The multi-objective optimisation problem can be stated as follows (Gupta *et al*., 1998):

$$\text{min (with respect to } q\text{)} \quad F(q) = \{f_1(q), \ldots, f_m(q)\}$$

where $f_i(q)$ are different **non-commensurable** measures of the goodness of fit and q is a particular parameter set.

A more mathematically rigorous definition of a Pareto optimal set, than the one given by Chankong and Haimes (1993) above, is stated in Gupta *et al*. (1998). Any Pareto set member q_i has the properties:

(1) For all nonmembers q_j there exists at least one member q_i such that $F(q_i)$ is strictly less than $F(q_j)$, and

(2) it is not possible to find q_j within the Pareto set such that $F(q_j)$ is strictly less than $F(q_i)$ (i.e., by "strictly less than" we mean $f_k(q_j) < f_k(q_i)$ for all k=1,…,m).

Or, in plain words, it is possible to divide the parameter space into 'good' and 'bad' solutions. However, one cannot favour one of the good solutions, since there is always another one that is better in a certain aspect, i.e. with respect to another objective function.

EXAMPLE

A simple example of Pareto optimisation is given in Goldberg (1989, p.199) using Schaffer's second function F_2 which is a function with two values $F_{21}(x)$ and $F_{22}(x)$. The task is now to find the values of the independent parameter x which minimises both functions.

$$F_{21}(x) = x^2$$

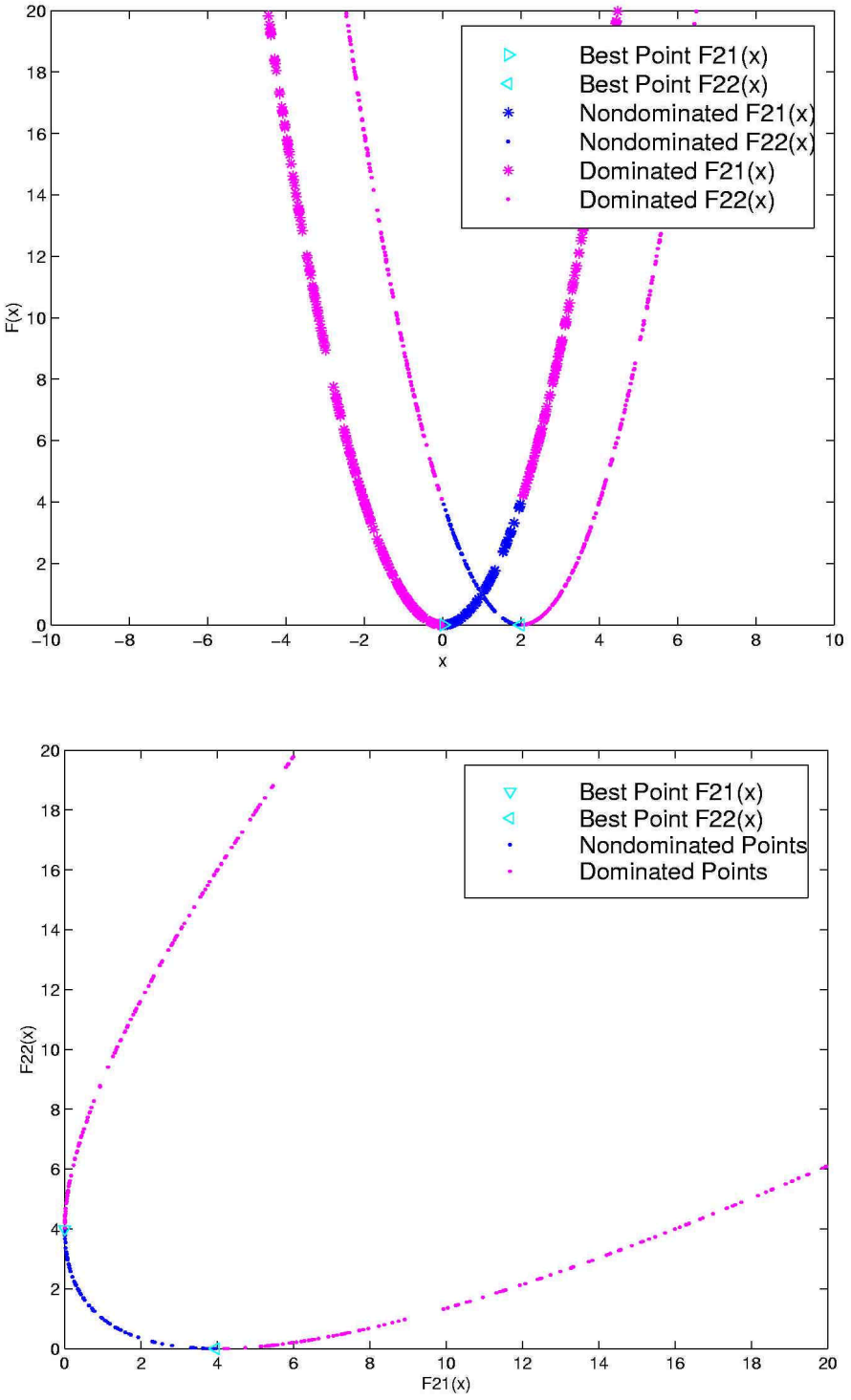


Figure 18 Optimisation results for Schaffer's second function.

$$F_{22}(x) = (x-2)^2$$

Sampling the feasible parameter space produces Figure 18. Figure 18-top shows the parameter range with the resulting function values. The function minima are at different places in the parameter space. In order to go

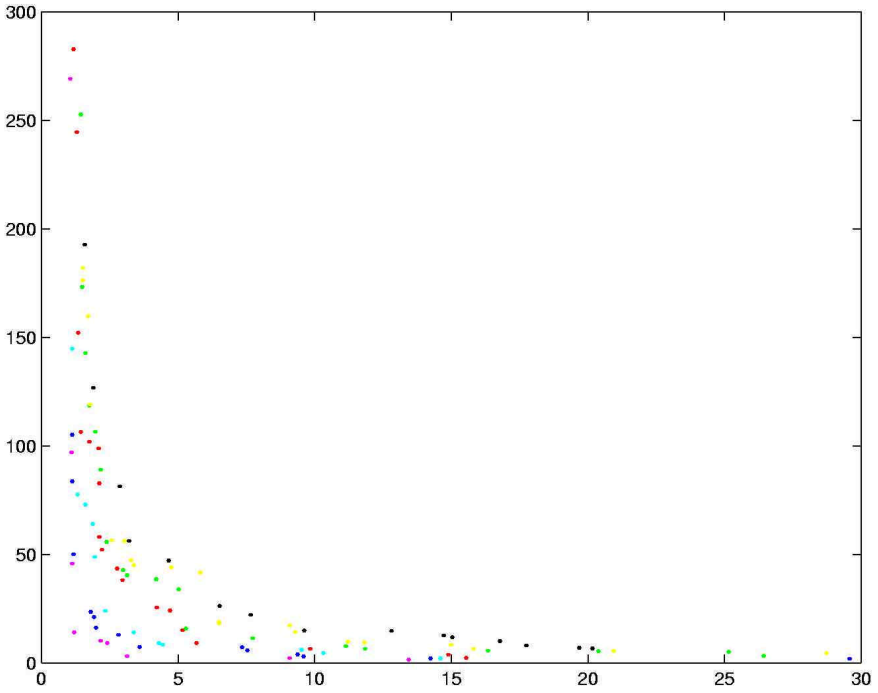


Figure 19 Different Pareto frontiers shown in different colours.

from one minimum to the other, one has to accept an increase in the result of the other function. The area between the two 'optimum' values (shown in blue) is called the **Pareto space** (Gupta *et al*., 1998).

Figure 18-bottom shows both function values plotted against each other. The blue dots, *i.e.* the Pareto set, is also called the **Pareto front**. The reason for that is obvious from the figure which reveals the trade off of one function against the other.

PARETO RANKING

A parameter population can be ranked based on nondomination, *i.e.* Pareto optimality (Goldberg, 1989, p.201). All nondominated parameter sets in the population are identified and assigned the rank one. The rank one set is then taken out of the population. The nondominated set for the remaining population is then identified. These parameter sets are assigned rank two. They are removed from the population and the procedure starts again. It continues till the whole population is ranked.

A visual illustration of a ranked parameter population is shown in Figure 19, which displays a number of different Pareto frontiers. All parameter sets with the same rank lie on the same Pareto frontier (Sorooshian *et al*., 1998).

Note that only nondominated values with rank one are identified by MCAT.

APPENDIX B

COMPARISON OF UNCERTAINTY ESTIMATION METHODS

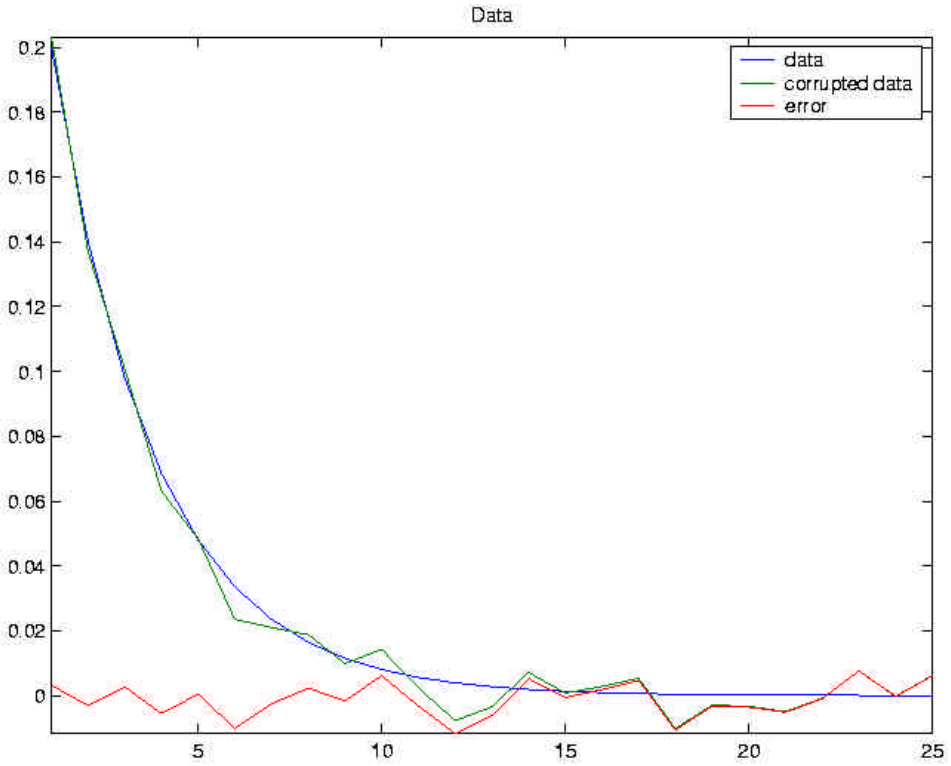


Figure 20 Synthetic data for the comparison example.

A script ('mcat_comp.m') that performs a comparison of different uncertainty estimation methods is included in the MCAT directory. Only the main figures of the example are provided here. Explanation is given while running through the example.

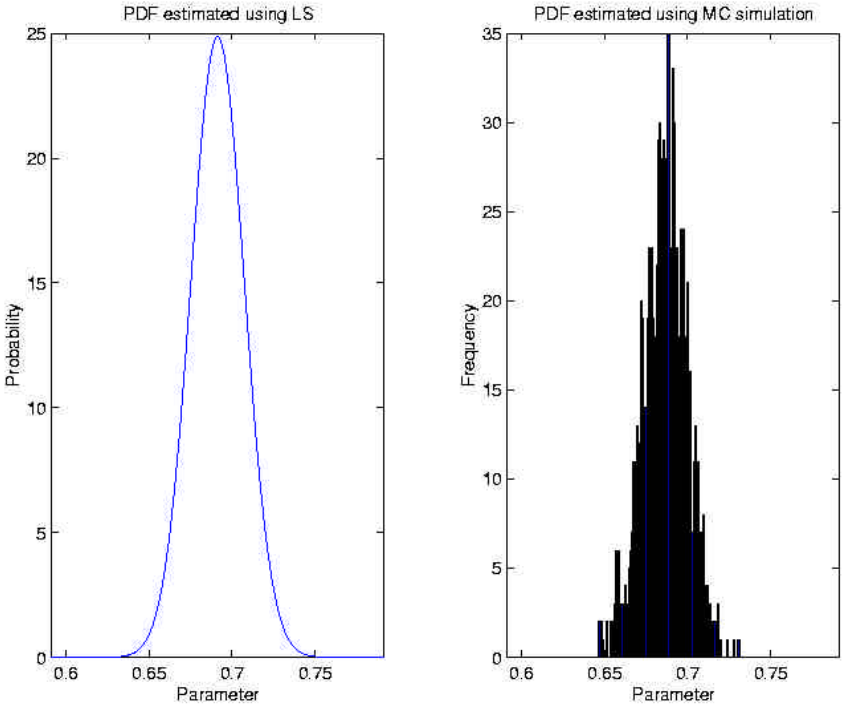


Figure 21 PDF's obtained using Least Squares and using Monte Carlo simulation.

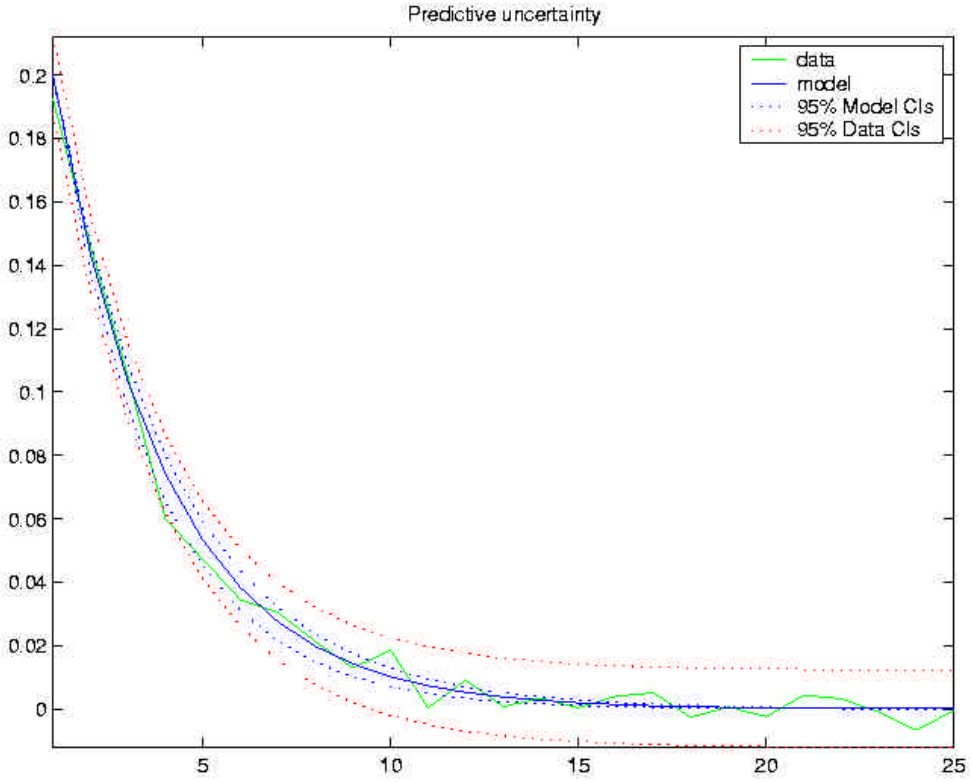


Figure 22 Model output confidence limits from the MC generated PDF and data confidence limits from the standard deviation of the residuals.

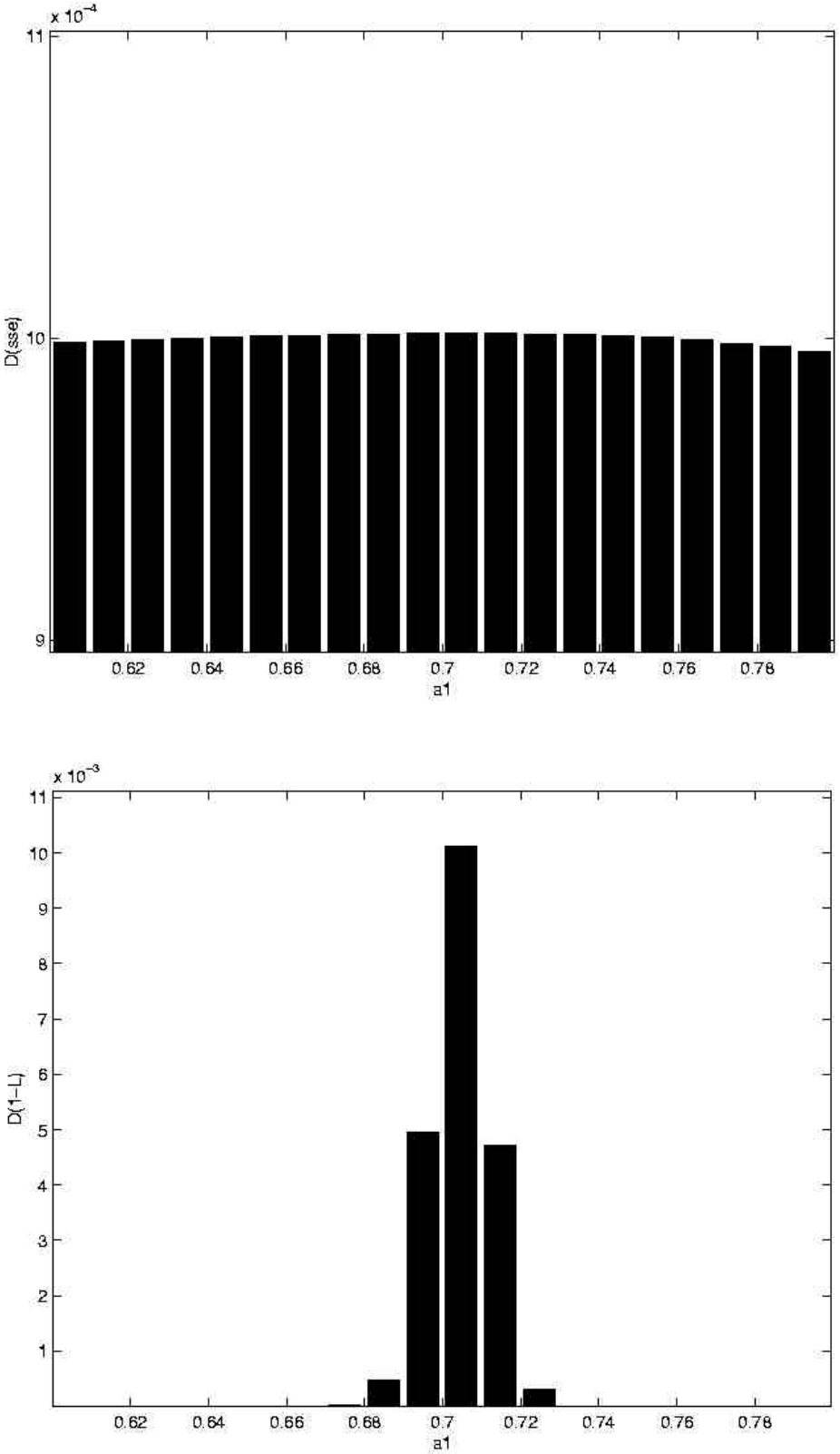


Figure 23 A posteriori parameter distributions using the sse (top) and the likelihood (bottom) objective function.

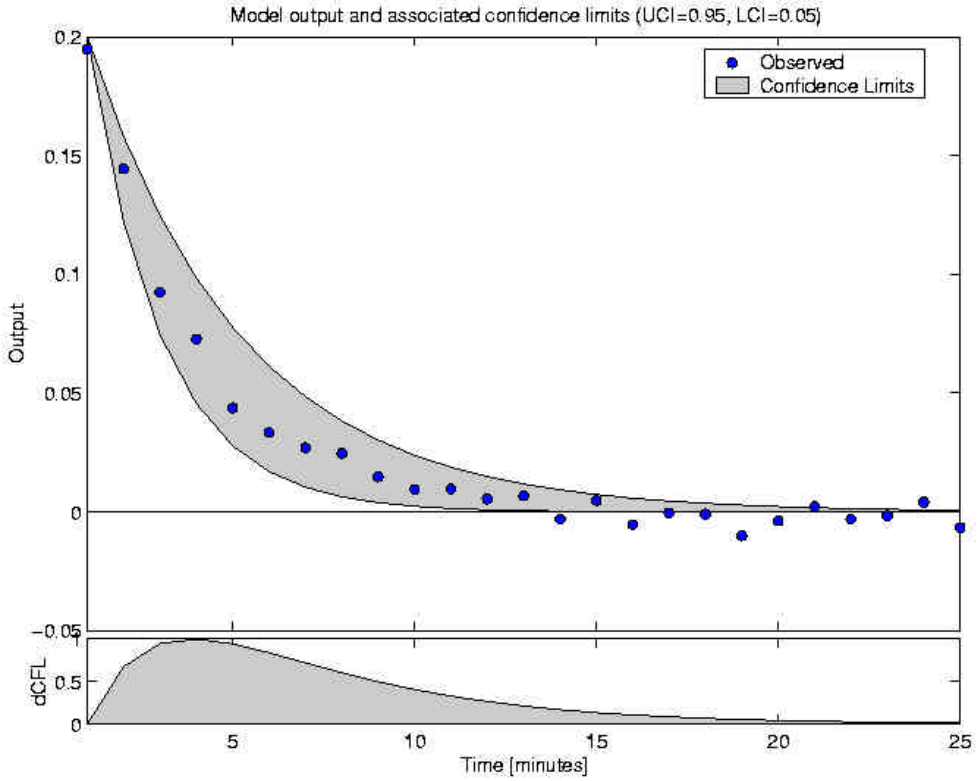


Figure 24 Uncertainty limits obtained calculated from the sse objective function in the GLUE approach without thresholding.

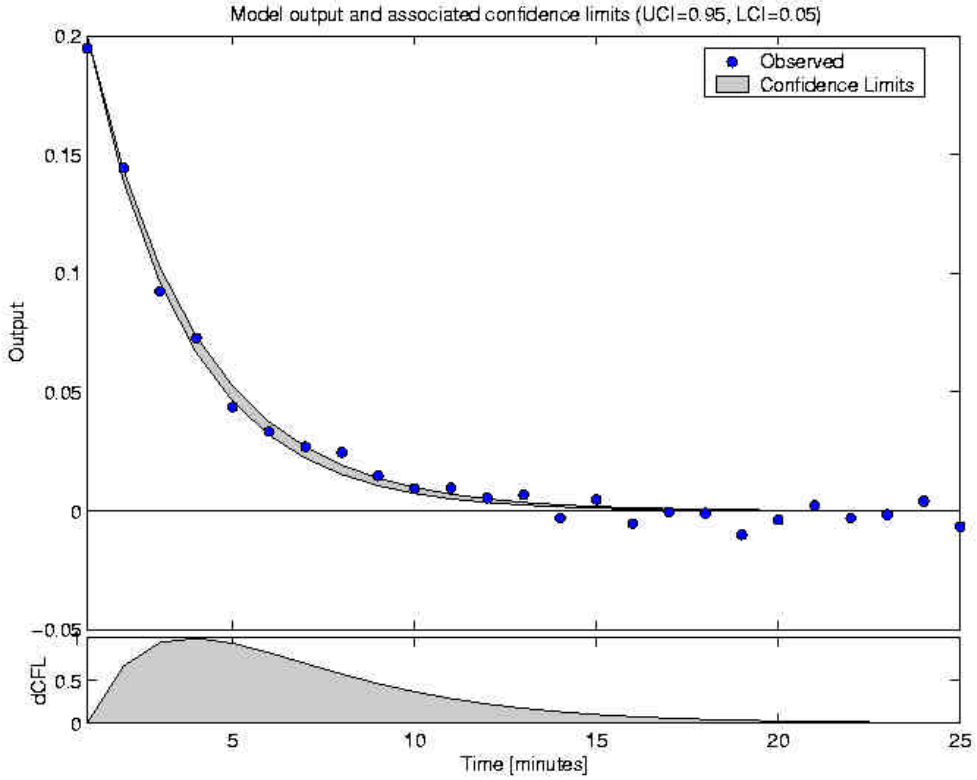


Figure 25 Uncertainty limits obtained calculated from the likelihood in the GLUE approach without thresholding.